CSc 3320: Systems Programming

Spring 2021

Homework

4: Total points 100

Submission instructions:

1. Create a Google doc for each homework assignment submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing in your document TWO POINTS WILL BE DEDUCTED per submission.
4. Keep this page 1 intact on all your submissions. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED per submission.
5. Each homework will typically have 2-3 PARTS, where each PART focuses on specific topic(s).
6. Start your responses to each PART on a new page.
7. If you are being asked to write code copy the code into a separate txt file and submit that as well.
8. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and copy the same into the document.
9. Upon completion, download a .PDF version of the document and submit the same.

Full Name: Maharshi Patel

Campus ID: mpatel185

Panther #: 002466568

ALL PROGRAMS MUST BE COMMENTED. YOUR SOLUTION WILL NOT BE ACCEPTED IF THERE ARE NO COMMENTS IN YOUR SCRIPT. Also note that the comments MUST be useful and not be random.

PART 1: 40pts
Must incorporate use of Functions and Pointers

1. Write a C program `checkPasswd.c` to check if the length of a given password string is 10 characters or not. If not, deduct 5 points per missing character. If the total deduction is greater than 30 points, print out the deduction and message "The password is unsafe! Please reset."; otherwise, print out "The password is safe."

```
#include <stdio.h>
#include <string.h>

int main() {

//initialize
	char chars;
	int i = 0;
	printf("Enter your password: ");

	  //do while loops and stores each character while it is not empty space (space bar)
	//also, increments i so it can be used for difference
	do {
		chars = getchar();
		i++;
	} while(chars != '\n');
```

```
        //stores value of password and compares with recommended
length
        int difference = 10 - i + 1;
        int score = -(difference * 5);
        printf("Score: %d\n",score);

        //if loops and determines if the password is safe
        if(score <= -30) {
                printf("The password is unsafe! Please reset.");
         }
        else {
                printf("The password is safe.");
         }
        return 0;
}
```

2. Similar to above question, update the C program `checkPasswd.c` to check if a password is safe or by not by checking only the evaluation criteria below. It will still print out the final score, and "safe" or "unsafe" when deduction is more than 30 points.

- Missing lower case -20 points
- Lack of capital letters -20 points
- Missing numbers -20 points
- More than 2 consecutive characters (e.g. 123 or abc) -20 points

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main() {

    //initialize
    int score=0, uppcount=0, lowcount=0, numcount=0;
   char chars;
   int i = 0;

    printf("Enter your password: ");

   //while loops stores each character while it is not empty space (space bar)
    while ((chars = getchar()) != '\n') {

        //iterates lowcount for every lower case character
        if(chars>='a' && chars<='z') {

            lowcount++;
```

```
            }

            //iterates uppcount for every uppercase character
            if(chars>='A' && chars<='Z') {
                  uppcount++;
            }

            //iterates numcount for every number character
            if(chars>='0' && chars<='9') {
                  numcount++;
            }

            //if loops will count the score according to the lowcount,
uppcount, and numcount
            if(lowcount==0){
                  score+=20;
            }
            if(uppcount==0){
                  score+=20;
            }
            if(numcount==0){
                  score+=20;
            }
      }

      //print the score
      printf("nScore = %d\n",(-score));

      //if loop checks whether password is safe, depending on the
score
```

```
        if(score>30){
                printf("The password is unsafe! Please reset.n");
        }
        else{
                printf("The password is safe.n");
        }
return 0;
}
```

Part II : 40pts
Must incorporate the use of Functions and Pointer arrays

3. Write a program that reads a message (can be characters, numeric or alphanumeric) and checks whether it is a palindrome (the characters in the message are the same when read from left-to-right or right-to-left).

```
#include <stdio.h>
#include<string.h>
#define MAX_LENGTH 100

int main() {
   //length of char array str is set MAX_LENGTH
   char str[MAX_LENGTH];

   //print and then scan word
   printf("Write a word for palindrome check: \n");
   scanf("%s",str);

   //checks whether str is palindrome
   if (palin(str,strlen(str))) {
      printf("%s is a palindrome\n",str);
   }
   else {
      printf("%s is not palindrome\n",str);
   }

   return 0;
}

int palin(char *str, int str_length) {
```

```
    //initialize
    int num = 1;
    char rev[MAX_LENGTH];
    int j = 0;

    //for loops and reassigns elements in the rev char array
    for (int i = str_length - 1; i >= 0; i--) {
        rev[j] = str[i];
        j++;
    }

    rev[j]='\0';

    //checks whether both arrays are not same
    for (int i = 0; i < str_length; i++) {
        if (str[i] != rev[i]) {
            num = 0;
            break;
        }
    }
    return num;
}
```

4. Write a program that will swap two variables without the use of any third variable. Utilize this program to write a program that reads two sentences that contain alphanumeric characters and the program must swap all the numerics in sentence1 with alphabet characters from sentence 2 and vice-versa. Keep the lengths of the sentences as identical.

```
#include <stdio.h>

int main() {
   char str1[20]= "1234abcd";
   char str2[20]= "abcd1234";

   //runs only if both strings are same sizes
   if(strlen(str1) == strlen(str2)) {

   //for loops until string length of sentence1 in met
   for(int i = 0; i < strlen(str1); i++) {

      //if loops check alphanumeric elements in both strings
      //and swaps numbers for letters and vice versa
      if(str1[i]>='0' && str1[i]<='9') {
         str1[i] = str2[i]-str1[i];
         str2[i] = str2[i]-str1[i];
         str1[i] = str1[i]+str2[i];
      }

      if(str2[i]>='a' && str2[i]<='z') {
         str2[i] = str1[i]-str2[i];
         str1[i] = str1[i]-str2[i];
         str2[i] = str2[i]+str1[i];
```

```
        }
    }

    }

    printf("sentence1: %s\n",str1);
    printf("sentence2: %s\n",str2);
    return 0;
}
```

Part III : 20pts
Must incorporate Functions, Pointers or PointerArrays, and Structures or Unions

5. Write a program that asks the user to enter an international dialing code and then looks it up in the country_codes array (see Sec 16.3 in C textbook). If it finds the code, the program should display the name of the corresponding country; if not, the program should print an error message. For demonstration purposes have at least 20 countries in your list.
   (Programming Project 1 on pg412 in C textbook)

```
#include <stdio.h>

//group both country name and the code num
struct dialing_code {
   char *country;
   int num;
};

int main(int argc, char* argv[]) {

   //initialize
   int input, i;

   //input names and code num in struct
   const struct dialing_code country_codes[] = {
         {"Argentina", 54}, {"Bangladesh", 880}, {"Brazil", 55},
{"Burma (Myanmar)", 95},
         {"China", 86}, {"Colombia", 57}, {"Congo,Dem.", 243},
{"Egypt", 20},
```

```
            {"Ethiopia", 251}, {"France", 33}, {"Germany", 49},
{"India", 91},
      {"Indonesia", 62}, {"Iran", 98}, {"Italy", 39}, {"Japan", 81},
             {"Mexico", 52}, {"Nigeria", 234}, {"Pakistan", 92},
{"Philippines", 63},
      {"United States", 1}
   };

   //assign value to entry
   int entry = sizeof(country_codes) / sizeof(*country_codes);

   //do while loops forever
   do {
      //initialize and print and then scan input code
      int found = 0;
      printf("Please input the international code(-1 to exit): ");
      scanf("%d", &input);

      //if loop to stop the code once input is -1.
      //it also stops the whole program
      if (input == -1){
         break;
      }

      //for loop until condition is met
      for (i = 0; i <= entry; i++) {

                  //if input value is one of the elements in
country_codes,
         //then print the country name using struct
```

```
            if (country_codes[i].num == input) {
                                        printf("The   country   is:   %s\n",
country_codes[i].country);
                found = 1;
            }
        }

        //if found is 0, then state that code not found
        if (!found) {
            printf("Code not found.\n");
        }
    } while(1);
    return 0;
}
```